EXHIBIT 5

IN THE UNITED STATES DISTRICT COURT
FOR THE NORTHERN DISTRICT OF CALIFORNIA
SAN FRANCISCO DIVISIONS

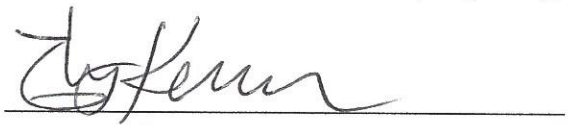
ORACLE AMERICA, INC.,	)	
	)	
Plaintiff,	)	
	)	
v.	)	Civil Action No. 10-03561 WHA
	)	
GOOGLE INC.,	)	
	)	
Defendant.	)	

EXPERT REPORT OF CHRIS F. KEMERER, Ph.D.

January 8, 2016

CONFIDENTIAL – ATTORNEYS' EYES ONLY
PURSUANT TO PROTECTIVE ORDER

I declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct. Executed on this 8th day of January, 2016, in Pittsburgh, PA.



Chris F. Kemerer, PhD
Chris F. Kemerer PhD LLC

TABLE OF CONTENTS

independently reproduce portions of the code and organization analysis above, and obtained the same results. Based on a review of these results, I conclude that the code and the structure, sequence and organization of the 37 Java API packages are reproduced in Android versions 2.3 and 5.1.

C. The Declaring Code and Structure, Sequence and Organization of the 37 Java API Packages in Java SE 5 were also copied in every version of Android between Android Versions 2.3 and 5.1.

44. I have conferred with Mr. Zeidman and believe his method of analysis and results regarding the code and structure, sequence and organization between Java SE 5 and the many Android versions between Versions 2.3 and 5.1 to be accurate. Mr. Zeidman reviewed the "diff reports" showing changes between different versions of the Android API, in order to determine whether any of the code and structure, sequence and organization of that code in the 37 Java API packages was listed on those "diff reports." Based on a review of the results in the Zeidman Report, Exs. Y, Z, AA, BB, CC, I conclude that code and structure, sequence and organization of the 37 Java API packages are reproduced in all Android release versions between 2.3 and 5.1.

D. Android Version 6 (Marshmallow, API Level 23)

45. As can also be seen in Mr. Zeidman's report, the same code and the same structure, sequence and organization of the 37 Java API packages are found in Android Version 6.0 (Marshmallow, API level 23). As detailed in Mr. Zeidman's report, the copying of the code and structure, sequence and organization of Java SE 5 code into Android Version 6.0, can be seen through evidence that the copied lines of code in Android Version 5.1 was also present in Android Version 6.0. (See Zeidman Report, Exs. S, T).

46. I have conferred with Mr. Zeidman and believe his method of analysis and results regarding the code and structure, sequence and organization between Java SE 5 and Android Version 6.0 to be accurate. Mr. Zeidman reviewed the "diff reports" showing changes between Android Version 5.1 and Version 6.0 in order to determine whether any of the code and structure, sequence and organization of that code in the 37 Java API packages was listed on those "diff reports." This process revealed that there were no changes in the copied code or structure, sequence and organization. (See Zeidman Report, Exs. Y, Z, AA, BB, CC). Based on a review of these results, I conclude that code and structure, sequence and organization of the 37 Java API packages are reproduced in Android Version 6.0.

E. Google Continued and Expanded Copying

47. As can also be seen in Mr. Zeidman's report, Google copied code and the structure, sequence and organization from new versions of Java, Java SE 6 and Java SE 7, into Android. As detailed in Mr. Zeidman's report, this can be seen through evidence that declaring code that was introduced in Java SE 6 and Java SE 7

was selectively added to the same classes in Android API levels 9, 11, 14, 19 and 21. (See Zeidman Report, Exs. V, W) Google copied from Java SE 6 and Java SE 7 into versions of Android that were released after the decision by the Federal Circuit Court of Appeals in this case. (See Zeidman Report, Exs. CC, Table 1, 4 and ¶ 121).

48. I have conferred with Mr. Zeidman and believe his method of analysis and results regarding the code and structure, sequence and organization between Java SE 6 and SE 7 and Android to be accurate. Based on a review of these results, I conclude that code and structure, sequence and organization of the 37 Java API packages from Java SE 6 and Java SE 7 are reproduced in Android.

49. The Court of Appeals for the Federal Circuit found that the declaring code and the structure, sequence and organization of the Java API packages are copyrightable. As explained by the Court of Appeals for the Federal Circuit, "the declaring code and the structure, sequence, and organization of the 37 Java API packages at issue are entitled to copyright protection."[23] The declaring code and organization of that code at issue in the Federal Circuit also appeared in the public facing API Specification. I have reviewed, with my research team, the code and organization of code in the classes from Java SE 6 and Java SE 7 discussed here. Like the code at issue in the Federal Circuit opinion, the copied code from Java SE 6 and Java SE 7 discussed here, identifies, introduces and specifies packages, classes, and methods, as well as other elements. For this reason, the code in Java SE 6 and Java SE 7 has the same expressive properties as the code in the classes already addressed by the Court. In other words, the code could have been written in multiple different ways, is creative and serves a similar purpose of declaring, for example, methods. On that basis, I conclude that the code from Java SE 6 and Java SE 7, addressed here, is copyrightable, for the reasons discussed by the Court of Appeals.

50. Similarly, like the code addressed by the Court of Appeals, the additional code from Java SE 6 and Java SE 7 is organized in an intricate hierarchy and elaborately organized taxonomy of expressions. On this basis, I conclude that the structure, sequence and organization of the code from Java SE 6 and Java SE 7 addressed here is copyrightable for the reasons discussed in the Court of Appeals opinion.

F. Google has Copied Implementing Code of the 37 Java API Packages.

51. As can also be seen in Mr. Zeidman's report, some of the code from the 37 Java API packages copied in Android constitutes declaring code from private classes, methods, variables and constructors. See Zeidman Report, ¶¶ 71-75. Private classes are comprised of code that does not occur in the public-facing Java API Specification. Copying of this code from private classes constitutes copying of a type of implementing code. Private classes can be used by developers to implement public API packages. For

[23] United States Court of Appeals for the Federal Circuit, Oracle America Inc. v. Google Inc. 2013-1021, -1022. p. 69.

For example, as discussed, Java enables applications to run in a variety of computing devices, independent of the idiosyncrasies of a given device's processor architecture.

185. Compatibility affords developers relative certainty that Java applications they develop will, in fact, consistently run across a variety of devices and that they will have access to a large community of users associated with such devices. Compatibility also reduces the costs of software development for such developers, in that they do not have to substantially invest in redeveloping software for a variety of underlying hardware or operating system contexts.

186. Compatibility also affords users relative certainty that Java applications that they acquire will run across a variety of devices. Thus Java compatibility enables availability of a greater number and variety of applications to users.

187. Java compatibility creates incentives for more developers to invest in writing programs for the Java platform and incentives for users to acquire and use such applications.

188. Oracle has taken steps to enforce compatibility within the Java ecosystem, and to maintain the "write once, run anywhere," proposition of Java, and to obtain the benefits discussed above, including those listed below.

- Sun and Oracle defined API specifications for particular versions of Java (for example, Java SE 5, SE 6 and SE 7), which define the set of API packages that are to be implemented and how they are to behave.

- Sun and Oracle maintain the Technology Compatibility Kit (TCK), which is a suite of tests against which licensed implementations of the Java API packages are to be tested for compatibility.

- Sun's and Oracle's agreements require that implementations of the Java platform conform to the set of APIs set forth in the specifications and prohibit implementations of the Java platform that include a subset or a superset of the API packages. For example, the "Specification License" that applies to the Java SE 5 API specification requires that any implementation of the Java SE 5 API specification: "(i) fully implements the Spec(s) including all its required interfaces and functionality; (ii) does not modify, subset, superset or otherwise extend the Licensor Name Space, or include any public or protected packages, classes, Java interfaces, fields or methods within the Licensor Name Space other than those required/authorized by the Specification or Specifications being implemented; and (iii) passes the TCK (including satisfying the requirements of the applicable TCK Users Guide) for such Specification. The foregoing license is expressly conditioned on your not acting outside its scope. No license is granted hereunder for any other purpose." TX 610.1. Sun and Oracle's commercial licenses also had these requirements. TX 498 ("may not subset or superset the Java Classes" and requiring "compatibility testing" against the "applicable TCK")

> DEX bytecode, meaning there's only one DEX bytecode incarnation of it. In fact, in the design -- the original design of Dalvik, it's -- it's specifically the case that that's, you know -- we could do optimization of DEX bytecode, so you could end up with multiple forms of DEX bytecode for a given thing. … Q So -- so the result of converting Java bytecode to DEX bytecode may be a situation in which there's not -- not a unique mapping between Java bytecode to DEX bytecode? A That's correct. Q Do you think it's possible that in the -- in the conversion of Java bytecode to DEX bytecode that some Java bytecode might just simply be ignored by -- by Android?... THE WITNESS: Well, there is -- there is bytecode that is ignored specifically by that compiler, because we don't support that bytecode. … Q Well, what do you mean, there's bytecode that is ignored specifically by which compiler? A By the DEX compiler that you're referring to." (Ghuloum 30(b)(6), Anwar, 188:15-192:12, Dec. 9, 2015)

206. On June 6, 2007, in an email from Andy Rubin, he recognized that: "[Sun] will never certify our VM, or put another way, I will not submit Dalvik for certification. Is it possible to certify without giving Sun access to the whole stack?" TX 246, GOOGLE-02-00089698-99.

207. As of December 2015, a high ranking Android executive with responsibilities for testing Android compatibility testified that he had never even looked at the Java TCK:

> "Q Do you have a role in dealing with compatibility issues -- compatibility issues in Android? A Yep. Q And notwithstanding that that's your role, you've never looked at the Java TCK? A Nope. Only look at the Java language specification. Q Okay. And so no -- anybody in your organization who deals with Android compatibility ever looked at or used the Java TCK? A Not that I'm aware of." (Ghuloum 30(b)(6), Anwar, 196:6-196:18, Dec. 9, 2015)

C. Google's Use of the 37 Java API Packages in Android Creates Fragmentation in the Java Developer Community

208. Google's unauthorized copying of the 37 API packages, its violation of the principles of no "subsetting" or "supersetting" in relation to the API packages that it copied from the official Java SE 5, Java SE 6 and Java SE 7 API specifications causes fragmentation to the Java platform and ecosystem. Google both subsetted and supersetted or otherwise included public or protected packages, classes, interfaces and fields other than those required or authorized by the Java API Specification at issue. I have discussed some examples in my report and Prof. Schmidt provides other examples in his report, which I incorporate and rely upon. (Schmidt Report, ¶ 105-107). Google also caused fragmentation by using the copied packages in compiled class libraries that are incompatible with Java, cause fragmentation to the Java platform and ecosystem. Fragmentation is a situation where (1) a Java developer writes an application for a particular Java specification (for example, a version of Java SE or Java ME), (2) expects to be able to use the API packages defined in the Java specification and expects their resultant application will run across devices that support Java according to the defined specification, but (3) then is faced with a Java-based platform that does not conform to the known and expected specifications.

> To put it bluntly: Android as it is currently defined is a fork of the Java ME platform. Android is similar to the Java ME, but it's a non-conformant implementation. Android is not compliant with Java ME nor is it compliant with Java SE. In fact, it's not really Java. Although it uses the Java programming language, the core APIs and the virtual machine are not consistent with the Java ME or SE platform - it's a fork. This was first pointed out by Stefano Mazzocchi in his November 12th blog entry entitled "Dalvik: how Google routed around Sun's IP-based licensing restrictions on Java ME". Stefano missed the fact that Android does not properly implement the CDC or CLDC Java ME APIs (a minimum requirement for Java ME conformance) - but kudos to him for being the first to report on the fork. The fork has since been picked up in the blogsphere by others here, here and elsewhere.

215. Other industry observers have noted the same. OAGOOGLE0000191668 (discussing "complaints about fragmentation" in Android, and observing that there would have been benefits if Google were "using/licensing" Java); OAGOOGL0004381807 ("it looks as though Google's Android is already beginning to fracture the Java mobile community."). Sun was concerned about Android because "API fragmentation means WORA goes out the window, which means the cost of exit goes way up and Sun's in trouble." OAGOOGLE0003997531-OAGOOGLE0003997532

216. There is evidence that Google was and continues to be aware that its incompatible use of the 37 Java APIs in Android fragmented Java. For example, on May 7, 2009, Andy Rubin referred to Android as a "fork" of Java. TX 172, GOOGLE-01-00029329.

D. The 37 API Packages in Android Are Technically Incompatible with the Java Platform

217. I have worked with Prof. Doug Schmidt who carried out analysis of and generated a report detailing numerous ways in which Android is incompatible with Java (in particular as relating to the 37 Java API packages). I rely upon and incorporate by reference in its entirety and refer to the contents of Prof. Schmidt's report. As detailed further in Prof. Schmidt's report, there are numerous reasons why Android is incompatible with Java. This technical incompatibility creates an incompatibility with Java developers' expectations that they can "write once, run anywhere" if they write programs targeted for a particular Java specification or platform.

218. The 37 Java API packages in Android do not pass the Java compatibility tests (the TCK), and thus do not meet any existing compatibility definition. (Schmidt Report, ¶¶ 98, 107). The Java SE 7 TCK test results generated 303 errors, resulting in failure of the test. (Schmidt Report, ¶¶ 104). The Java SE 6 TCK test results generated 409 errors, resulting in failure of that test. (Schmidt Report, ¶¶ 103).

APPENDIX G – Evolution of Java API Packages

Java API Packages JDK Versions 1-8

Package	1.0	1.1	1.2	1.3	1.4	1.5	6.0	7.0	8.0
sun.tools.debug	X								
java.awt.peer	X								
java.applet	X	X	X	X	X	X	X	X	X
java.awt.image	X	X	X	X	X	X	X	X	X
java.awt	X	X	X	X	X	X	X	X	X
java.net	X	X	X	X	X	X	X	X	X
java.util	X	X	X	X	X	X	X	X	X
java.io	X	X	X	X	X	X	X	X	X
java.lang	X	X	X	X	X	X	X	X	X
java.awt.datatransfer		X	X	X	X	X	X	X	X
java.awt.event		X	X	X	X	X	X	X	X
java.beans		X	X	X	X	X	X	X	X
java.lang.reflect		X	X	X	X	X	X	X	X
java.math		X	X	X	X	X	X	X	X
java.rmi		X	X	X	X	X	X	X	X
java.rmi.dgc		X	X	X	X	X	X	X	X
java.rmi.registry		X	X	X	X	X	X	X	X
java.rmi.server		X	X	X	X	X	X	X	X
java.security		X	X	X	X	X	X	X	X
java.security.acl		X	X	X	X	X	X	X	X
java.security.interfaces		X	X	X	X	X	X	X	X
java.sql		X	X	X	X	X	X	X	X
java.text		X	X	X	X	X	X	X	X
java.util.zip		X	X	X	X	X	X	X	X
java.awt.color			X	X	X	X	X	X	X
java.awt.dnd			X	X	X	X	X	X	X
java.awt.font			X	X	X	X	X	X	X
java.awt.geom			X	X	X	X	X	X	X
java.awt.im			X	X	X	X	X	X	X
java.awt.image.renderable			X	X	X	X	X	X	X
java.awt.print			X	X	X	X	X	X	X
java.beans.beancontext			X	X	X	X	X	X	X
java.lang.ref			X	X	X	X	X	X	X
java.rmi.activation			X	X	X	X	X	X	X
java.security.cert			X	X	X	X	X	X	X
java.security.spec			X	X	X	X	X	X	X
java.util.jar			X	X	X	X	X	X	X
javax.accessibility			X	X	X	X	X	X	X
javax.swing			X	X	X	X	X	X	X

Package	1.0	1.1	1.2	1.3	1.4	1.5	6.0	7.0	8.0
javax.swing.border			X	X	X	X	X	X	X
javax.swing.colorchooser			X	X	X	X	X	X	X
javax.swing.event			X	X	X	X	X	X	X
javax.swing.filechooser			X	X	X	X	X	X	X
javax.swing.plaf			X	X	X	X	X	X	X
javax.swing.plaf.basic			X	X	X	X	X	X	X
javax.swing.plaf.metal			X	X	X	X	X	X	X
javax.swing.plaf.multi			X	X	X	X	X	X	X
javax.swing.table			X	X	X	X	X	X	X
javax.swing.text			X	X	X	X	X	X	X
javax.swing.text.html			X	X	X	X	X	X	X
javax.swing.text.html.parser			X	X	X	X	X	X	X
javax.swing.text.rtf			X	X	X	X	X	X	X
javax.swing.tree			X	X	X	X	X	X	X
javax.swing.undo			X	X	X	X	X	X	X
org.omg.CORBA			X	X	X	X	X	X	X
org.omg.CORBA.DynAnyPackage			X	X	X	X	X	X	X
org.omg.CORBA.ORBPackage			X	X	X	X	X	X	X
org.omg.CORBA.portable			X	X	X	X	X	X	X
org.omg.CORBA.TypeCodePackage			X	X	X	X	X	X	X
org.omg.CosNaming			X	X	X	X	X	X	X
org.omg.CosNaming.NamingContextPackage			X	X	X	X	X	X	X
java.awt.im.spi				X	X	X	X	X	X
javax.naming				X	X	X	X	X	X
javax.naming.directory				X	X	X	X	X	X
javax.naming.event				X	X	X	X	X	X
javax.naming.ldap				X	X	X	X	X	X
javax.naming.spi				X	X	X	X	X	X
javax.rmi				X	X	X	X	X	X
javax.rmi.CORBA				X	X	X	X	X	X
javax.sound.midi				X	X	X	X	X	X
javax.sound.midi.spi				X	X	X	X	X	X
javax.sound.sampled				X	X	X	X	X	X
javax.sound.sampled.spi				X	X	X	X	X	X
javax.transaction				X	X	X	X	X	X
org.omg.CORBA_2_3				X	X	X	X	X	X
org.omg.CORBA_2_3.portable				X	X	X	X	X	X
org.omg.SendingContext				X	X	X	X	X	X
org.omg.stub.java.rmi				X	X	X	X	X	X
java.nio					X	X	X	X	X
java.nio.channels					X	X	X	X	X

Package	1.0	1.1	1.2	1.3	1.4	1.5	6.0	7.0	8.0
java.nio.channels.spi					X	X	X	X	X
java.nio.charset					X	X	X	X	X
java.nio.charset.spi					X	X	X	X	X
java.util.logging					X	X	X	X	X
java.util.prefs					X	X	X	X	X
java.util.regex					X	X	X	X	X
javax.crypto					X	X	X	X	X
javax.crypto.interfaces					X	X	X	X	X
javax.crypto.spec					X	X	X	X	X
javax.imageio					X	X	X	X	X
javax.imageio.event					X	X	X	X	X
javax.imageio.metadata					X	X	X	X	X
javax.imageio.plugins.jpeg					X	X	X	X	X
javax.imageio.spi					X	X	X	X	X
javax.imageio.stream					X	X	X	X	X
javax.net					X	X	X	X	X
javax.net.ssl					X	X	X	X	X
javax.print					X	X	X	X	X
javax.print.attribute					X	X	X	X	X
javax.print.attribute.standard					X	X	X	X	X
javax.print.event					X	X	X	X	X
javax.security.auth					X	X	X	X	X
javax.security.auth.callback					X	X	X	X	X
javax.security.auth.kerberos					X	X	X	X	X
javax.security.auth.login					X	X	X	X	X
javax.security.auth.spi					X	X	X	X	X
javax.security.auth.x500					X	X	X	X	X
javax.security.cert					X	X	X	X	X
javax.sql					X	X	X	X	X
javax.transaction.xa					X	X	X	X	X
javax.xml.parsers					X	X	X	X	X
javax.xml.transform					X	X	X	X	X
javax.xml.transform.dom					X	X	X	X	X
javax.xml.transform.sax					X	X	X	X	X
javax.xml.transform.stream					X	X	X	X	X
org.ietf.jgss					X	X	X	X	X
org.omg.CosNaming.NamingContextExtPackage					X	X	X	X	X
org.omg.Dynamic					X	X	X	X	X
org.omg.DynamicAny					X	X	X	X	X
org.omg.DynamicAny.DynAnyFactoryPackage					X	X	X	X	X

Package	1.0	1.1	1.2	1.3	1.4	1.5	6.0	7.0	8.0
org.omg.DynamicAny.DynAnyPackage					X	X	X	X	X
org.omg.IOP					X	X	X	X	X
org.omg.IOP.CodecFactoryPackage					X	X	X	X	X
org.omg.IOP.CodecPackage					X	X	X	X	X
org.omg.Messaging					X	X	X	X	X
org.omg.PortableInterceptor					X	X	X	X	X
org.omg.PortableInterceptor.ORBInitInfo Package					X	X	X	X	X
org.omg.PortableServer					X	X	X	X	X
org.omg.PortableServer.CurrentPackage					X	X	X	X	X
org.omg.PortableServer.POAManagerPac kage					X	X	X	X	X
org.omg.PortableServer.POAPackage					X	X	X	X	X
org.omg.PortableServer.portable					X	X	X	X	X
org.omg.PortableServer.ServantLocatorP ackage					X	X	X	X	X
org.w3c.dom					X	X	X	X	X
org.xml.sax					X	X	X	X	X
org.xml.sax.ext					X	X	X	X	X
org.xml.sax.helpers					X	X	X	X	X
java.lang.annotation						X	X	X	X
java.lang.instrument						X	X	X	X
java.lang.management						X	X	X	X
java.util.concurrent						X	X	X	X
java.util.concurrent.atomic						X	X	X	X
java.util.concurrent.locks						X	X	X	X
javax.imageio.plugins.bmp						X	X	X	X
javax.management						X	X	X	X
javax.management.loading						X	X	X	X
javax.management.modelmbean						X	X	X	X
javax.management.monitor						X	X	X	X
javax.management.openmbean						X	X	X	X
javax.management.relation						X	X	X	X
javax.management.remote						X	X	X	X
javax.management.remote.rmi						X	X	X	X
javax.management.timer						X	X	X	X
javax.rmi.ssl						X	X	X	X
javax.security.sasl						X	X	X	X
javax.sql.rowset						X	X	X	X
javax.sql.rowset.serial						X	X	X	X
javax.sql.rowset.spi						X	X	X	X
javax.swing.plaf.synth						X	X	X	X
javax.xml						X	X	X	X

Package	1.0	1.1	1.2	1.3	1.4	1.5	6.0	7.0	8.0
javax.xml.datatype						X	X	X	X
javax.xml.namespace						X	X	X	X
javax.xml.validation						X	X	X	X
javax.xml.xpath						X	X	X	X
org.w3c.dom.bootstrap						X	X	X	X
org.w3c.dom.events						X	X	X	X
org.w3c.dom.ls						X	X	X	X
javax.activity						X	X	X	X
java.text.spi							X	X	X
java.util.spi							X	X	X
javax.activation							X	X	X
javax.annotation							X	X	X
javax.annotation.processing							X	X	X
javax.jws							X	X	X
javax.jws.soap							X	X	X
javax.lang.model							X	X	X
javax.lang.model.element							X	X	X
javax.lang.model.type							X	X	X
javax.lang.model.util							X	X	X
javax.script							X	X	X
javax.tools							X	X	X
javax.xml.bind							X	X	X
javax.xml.bind.annotation							X	X	X
javax.xml.bind.annotation.adapters							X	X	X
javax.xml.bind.attachment							X	X	X
javax.xml.bind.helpers							X	X	X
javax.xml.bind.util							X	X	X
javax.xml.crypto							X	X	X
javax.xml.crypto.dom							X	X	X
javax.xml.crypto.dsig							X	X	X
javax.xml.crypto.dsig.dom							X	X	X
javax.xml.crypto.dsig.keyinfo							X	X	X
javax.xml.crypto.dsig.spec							X	X	X
javax.xml.soap							X	X	X
javax.xml.stream							X	X	X
javax.xml.stream.events							X	X	X
javax.xml.stream.util							X	X	X
javax.xml.transform.stax							X	X	X
javax.xml.ws							X	X	X
javax.xml.ws.handler							X	X	X
javax.xml.ws.handler.soap							X	X	X
javax.xml.ws.http							X	X	X

Package	1.0	1.1	1.2	1.3	1.4	1.5	6.0	7.0	8.0
javax.xml.ws.soap							X	X	X
javax.xml.ws.spi							X	X	X
javax.xml.ws.wsaddressing							X	X	X
java.lang.invoke								X	X
java.nio.file								X	X
java.nio.file.attribute								X	X
java.nio.file.spi								X	X
javax.swing.plaf.nimbus								X	X
javax.xml.ws.spi.http								X	X
java.time									X
java.time.chrono									X
java.time.format									X
java.time.temporal									X
java.time.zone									X
java.util.function									X
java.util.stream									X
org.w3c.dom.views									X
Total	9	22	59	76	135	166	203	209	217